1 grudnia 2019

Michał Wdowski
Grupa G5b

Metody Numeryczne
Sprawozdanie z Projketu 1

1 Treść zadania

23. Obliczanie wskaźnika uwarunkowania $cond(A)$ macierzy nieosobliwej $A \in \mathbb{R}^{n \times n}$:

$$cond(A) = \|A^{-1}\|_1 \cdot \|A\|_1.$$

Zaimplementować metodę eliminacji Gaussa z pełnym wyborem elementu głównego do wyznaczenia macierzy A^{-1}.

2 Metoda rozwiązania

Metoda Gaussa z pełnym wyborem elementu głównego (*GECP*) jest modyfikacją zwykłej metody eliminacji Gaussa (*GE*). Działanie *GE* polega na tym, że dla każdego $i = 1, \cdots, n-1$, dla każdego $j = i, \cdots, n$ wykonujemy eliminację na wierszach, czyli wierszowi a_j przypisujemy wartość $a_j - \frac{a_{i,j}}{a_{i,i}} \cdot a_i$. Sprawia to, że elementy w kolumnie i pod elementem $a_{i,i}$ są zerami. Jednocześnie w macierzy jednostkowej o tych samych wymiarach zapisujemy współczynniki eliminacji, czyli używane wcześniej $\frac{a_{i,j}}{a_{i,i}}$.

Modyfikacja w *GECP* polega na tym, że na początku każdego kroku i następuje wybór tzw. elementu wiodącego, czyli największego co do modułu elementu z podmacierzy $A^{(i \cdots n) \times (i \cdots n)}$, a następnie zamiana wierszy i kolumn w taki sposób, by element wiodący znalazł się na miejscu $i \times i$.

Odwracanie macierzy trójkątnych dolnych i górnych za pomocą metody Gaussa odbywa się w następujący sposób: do macierzy X dopisujemy macierz jednostkową. Następnie sprowadzamy X do macierzy jednostkowej poprzez operacje na wierszach, wykonując te same operacje na dopisanej macierzy I. Z układu $[X|I]$ doszliśmy do $[I|Y]$, i okazuje się, że $Y = X^{-1}$.

3 Sposób działania programu

Używając zadanej metody *GECP* funkcja `my_gecp(A)` program znajduje rozkład $PAQ = LU$ taki, że:

- P i Q - macierze permutacji wierszy i kolumn,
- L - macierz trójkątna dolna z jedynkami na przekątnej,
- U - macierz trójkątna górna.

Zauważmy, że $PAQ = LU$ jest równoważne zapisowi $A^{-1} = QU^{-1}L^{-1}P$. Oznacza to, że wystarczy odwrócić macierze L i U (funkcje `inverse_L(L)` i `inverse_U(U)`), a następnie wymnożyć te macierze.
Gdy mamy już A^{-1} (`inverse_A(A)`), należy znaleźć normę kolumnową macierzy A i jej odwrotności (`my_colnorm(A)`), czyli ich największe sumy wartości bezwzględnych elementów w każdej z kolumn.
Na końcu funkcja `my_cond(A)`, zwraca wynik zadany przez treść zadania - docelowo identyczny z `cond(A, 1)`.

4 Przykłady

Funkcja `my_cond.m` jest testowana na różnych rozmiarów (od 5 do 1000 wierszy i kolumn) macierzy losowych, macierzy Hilberta i macierzy z funkcji `gallery('lehmer')`.

Dla każdego przypadku w skrypcie `testy.m` obliczane są wyniki działania funkcji `my_cond(A)` oraz `cond(A, 1)`, a także różnica tych wyników; czasy ich działania i ich różnica: r_R i r_R, wyliczone przy odwracaniu macierzy A. Następnie w skrypcie `tabele.m` liczone są błędny względne i dane są porządkowane.
Do przekształcenia macierzy z MATLAB do formatu `.tex` użyłem skryptu `matrix2latex.m`, który napisał Moritz Koehler. Kod i licencja zostały załączone do kodów projektu.

Uwaga: skrót "b.w." oznacza "błąd względny", a czas podawany jest w sekundach

Tabela 1: Wyniki dla macierzy losowych

Wym.	`my_cond(A)`	`cond(A, 1)`	b.w. `my_cond(A)`	czas `my_cond(A)`	czas `cond(A, 1)`	b.w. czasów	r_R	r_L
5	226.6786	226.6786	6.2692e-16	0.030053	0.003519	7.5402	7.5884e-17	3.2194e-17
10	1154.8109	1154.8109	9.8446e-16	0.001402	7.2e-05	18.4722	6.1138e-17	5.7368e-17
15	479.1358	479.1358	3.7964e-15	0.000518	3.9e-05	12.2821	3.1702e-17	6.0889e-17
20	780.9473	780.9473	1.4558e-15	0.000707	0.000511	0.38356	6.9002e-17	2.7634e-17
25	882.749	882.749	0	0.001249	0.000631	0.9794	5.9294e-17	5.6445e-17
30	1040.1021	1040.1021	3.4977e-15	0.001193	0.000892	0.33744	1.2803e-16	5.8501e-17
50	2627.6608	2627.6608	4.3265e-15	0.00286	0.000108	25.4815	7.5597e-17	4.51e-17
75	2013.5371	2013.5371	9.1467e-15	0.007068	0.002393	1.9536	8.8715e-17	1.0636e-16
100	11111.4489	11111.4489	1.3587e-14	0.010426	0.000976	9.6824	6.3103e-17	4.819e-17
150	9391.4236	9391.4236	1.9369e-15	0.054731	0.009796	4.5871	5.9551e-17	7.3365e-17
200	40158.4013	40158.4013	3.7505e-14	0.10371	0.000625	164.9312	9.5892e-17	5.4306e-17
250	88726.6032	88726.6032	1.1792e-13	0.10047	0.001	99.467	8.442e-17	1.1921e-16
300	778390.8723	778390.8723	9.8455e-13	0.15767	0.001171	133.6422	6.3477e-17	7.639e-17
400	64489.6271	64489.6271	1.0583e-13	0.36335	0.001796	201.309	1.0548e-16	9.0479e-17
500	471195.4813	471195.4813	1.1978e-12	0.61917	0.002868	214.8888	1.2791e-16	7.3313e-17
600	79263.6062	79263.6062	4.2409e-14	1.0122	0.008884	112.934	1.5912e-16	1.2027e-16
700	42918.2073	42918.2073	2.1361e-14	1.786	0.006099	291.836	2.4223e-16	3.0018e-16
800	151915.9856	151915.9856	3.4484e-15	2.7883	0.00854	325.4958	1.1328e-16	1.2451e-16
900	634161.3956	634161.3956	3.7064e-13	4.5425	0.013845	327.0946	2.5963e-16	1.0479e-16
1000	258274.4397	258274.4397	1.9799e-13	5.6023	0.017224	324.2584	1.8232e-16	9.42e-17

.

Wyniki obliczania współczynnika uwarunkowania niemalże nie odbiegają od siebie - błąd wzglęgny jest bardzo mały. Widać jednak, że działanie funkcji MATLABowej jest dużo szybsze.

Tabela 2: Wyniki dla macierzy Hilberta

Wym.	`my_cond(A)`	`cond(A, 1)`	b.w. `my_cond(A)`	czas `my_cond(A)`	czas `cond(A, 1)`	b.w. czasów	r_R	r_L
5	943656	943656	4.8125e-13	0.00829	0.000757	9.9511	1.8034e-17	1.7358e-17
10	3.53538e+14	3.53523e+14	4.2107e-05	0.004284	0.000197	20.7462	2.3863e-17	2.1487e-17
15	1.10093e+18	1.20924e+18	0.089564	0.000434	0.000245	0.77143	4.7757e-17	1.6117e-17
20	2.34526e+19	1.04795e+19	1.2379	0.000605	0.000225	1.6889	3.3531e-17	3.0215e-17
25	2.07646e+19	1.73572e+19	0.19631	0.001255	0.000511	1.456	2.5374e-17	5.0509e-17
30	3.74342e+19	5.59337e+18	5.6926	0.001412	0.000467	2.0236	1.9881e-17	7.1831e-17
50	1.44962e+20	3.21565e+19	3.508	0.006674	0.000311	20.4598	8.4356e-17	2.0381e-17
75	1.25756e+20	2.12100e+20	0.40709	0.006826	0.000335	19.3761	6.3545e-17	5.2265e-17
100	1.79939e+20	2.08655e+21	0.91376	0.010006	0.000451	21.1863	3.9772e-17	2.5306e-17
150	3.72037e+20	5.55342e+20	0.33008	0.04338	0.000703	60.707	3.8809e-17	3.2694e-17
200	3.98503e+20	6.92506e+20	0.42455	0.068183	0.001025	65.52	5.0257e-17	5.4765e-17
250	5.84930e+20	1.51833e+21	0.61476	0.10743	0.001391	76.2293	4.32e-17	4.2848e-17
300	3.98835e+21	3.42437e+20	10.647	0.14218	0.001746	80.4324	7.2779e-17	3.4217e-17
400	8.49319e+20	9.84821e+20	0.13759	0.34716	0.002651	129.954	7.6985e-17	7.5184e-17
500	4.41617e+21	1.34863e+22	0.67255	0.63462	0.003857	163.5377	4.6449e-17	8.2937e-17
600	3.26791e+22	3.37741e+21	8.6758	0.97205	0.006311	153.0249	4.0718e-17	4.4776e-17
700	3.21559e+21	1.71660e+21	0.87323	1.7322	0.008224	209.6274	8.2301e-17	5.9592e-17
800	5.59125e+22	5.67497e+23	0.90148	2.8909	0.012978	221.753	1.5937e-16	6.8261e-17
900	5.62476e+22	7.59416e+21	6.4067	4.426	0.014466	304.9608	2.2166e-16	4.9862e-17
1000	1.81475e+22	2.15101e+22	0.15633	5.4985	0.016006	342.5286	8.2701e-17	4.7985e-17

W tych macierzach błąd względny wyników jest bardzo duży, co ciekawe - zwłaszcza dla wymiarów 30, 300, 600 i 900. Być może to, że te liczby są podzielne przez 3, dodatkowo wpływa na dokładność w odwracaniu macierzy. Macierze Hilberta same w sobie są źle uwarunkowane - na tyle, że przy uruchamianu funkcji `cond(A, 1)` MATLAB wypisuje ostrzeżenie o niedokładności wyniku.

Tabela 3: Wyniki dla macierzy z `gallery('lehmer')`

Wym.	`my_cond(A)`	`cond(A, 1)`	b.w. `my_cond(A)`	czas `my_cond(A)`	czas `cond(A, 1)`	b.w. czasów	r_R	r_L
5	26.8	26.8	2.6513e-16	0.002436	8.1e-05	29.0741	4.3275e-17	4.797e-17
10	114.7286	114.7286	3.716e-16	0.000348	3.9e-05	7.9231	3.0018e-17	3.5746e-17
15	263.2937	263.2937	3.2384e-15	0.000452	3.7e-05	11.2162	5.9627e-17	2.2783e-17
20	472.5052	472.5052	6.7369e-15	0.000627	4.1e-05	14.2927	5.6287e-17	3.9073e-17
25	742.365	742.365	2.1593e-14	0.001326	0.000298	3.4497	3.3753e-17	3.8152e-17
30	1072.8737	1072.8737	1.9074e-14	0.001186	7.7e-05	14.4026	3.5292e-17	5.7495e-17
50	3001.8149	3001.8149	6.4081e-14	0.002588	6.5e-05	38.8154	6.3353e-17	3.6936e-17
75	6777.6224	6777.6224	1.825e-13	0.00582	0.000103	55.5049	6.2403e-17	3.5159e-17
100	12069.6708	12069.6708	3.9184e-14	0.01036	0.000493	20.0142	5.3242e-17	4.1035e-17
150	27202.4856	27202.4856	7.7567e-15	0.052917	0.000479	109.4739	5.1717e-17	4.809e-17
200	48400.6129	48400.6129	1.8024e-13	0.057541	0.000686	82.879	6.4665e-17	4.1714e-17
250	75664.4098	75664.4098	3.2502e-13	0.099009	0.001017	96.354	5.9467e-17	4.2653e-17
300	108993.164	108993.164	9.7624e-13	0.18213	0.001493	120.9886	5.4368e-17	4.7721e-17
400	193846.9046	193846.9046	2.1772e-12	0.3356	0.002764	120.4175	6.0101e-17	5.1933e-17
500	302961.4763	302961.4763	3.7976e-12	0.59056	0.003205	183.263	6.0249e-17	4.9927e-17
600	436337.7939	436337.7939	7.7787e-12	1.0719	0.004785	223.0161	6.3634e-17	4.8616e-17
700	593975.1673	593975.1673	1.0377e-11	1.782	0.006425	276.3494	6.0782e-17	5.2554e-17
800	775873.3711	775873.3711	1.4039e-11	3.0826	0.010296	298.4021	6.7524e-17	5.3556e-17
900	982033.4556	982033.4557	1.871e-11	4.6122	0.016064	286.1157	6.83e-17	4.9197e-17
1000	1212454.4616	1212454.4617	2.8272e-11	5.2394	0.016088	324.6684	6.0561e-17	5.428e-17

W tym wypadku wyniki nie odbiegają zbytnio od wyników dla macierzy losowych.

5 Analiza wyników

Czas działania autorskiej funkcji do wyliczania wskaźnika uwarunkowania jest dużo większy dla większych macierzy - już dla rozmiarów 1000×1000 zajmuje mu to 5 sekund, co jest ponad 300 razy wolniejsze od domyślnej MATLABowej funkcji.
Warto zauważyć, że dla macierzy losowych błąd względny otrzymanych wyników był pomijalnie mały - oznacza to, że zadany algorytm został zaimplementowany w poprawny sposób.